6.897: CSCI8980 Algorithmic Techniques for Big Data September 19, 2013

Lecture 5

Dr. Barna Saha *Scribe: Shuo Zhou*

Overview

In the previous lecture we saw how using *sketching* the second frequency moment can be approximated well in space logarithmic in the input size. In this lecture, we explore the dimensionality reduction technique-a by-product of sketching method in the streaming model. We introduce Johnson-Linderstrauss (JL) Lemma, a very powerful tool for l_2-dimensionality reduction. JLlemma has found many applications in computer science. As a concrete application of JL lemma, we consider *nearest neighbor problem.* We introduce a popular technique of *locality sensitive hashing* (LSH) for the nearest neighbor problem. The discussion on LSH will continue in the next lecture as well.

1 Linear Sketch as Dimensionality Reduction Technique

The algorithm for estimating F_2 in the previous lecture maintains a linear sketch $[Z_1, Z_2, ..., Z_t] = R\mathbf{x}$ where R is a $t \times n$ random matrix with entries $\{+1, -1\}$ and x is the frequency vector. We Use $Y = ||Rx||_2^2$ to estimate $t||x|_2^2$. $t = O(\frac{1}{\epsilon^2})$. Then by scaling by a factor of t, we obtain an estimate of $F_2(x) = ||x||_2^2$. We have

$$\mathsf{Pr}\big[|Y/t - ||x||_2^2|| \leq \epsilon||x||_2^2\big] > \frac{1}{c}$$

where c is some constant. Streaming algorithm operating in the sketch model can be viewed as **dimensionality reduction** technique. From a stream which can be viewed as a n-dimensional vector, we obtain a sketch Y of dimension t. However, the distribution of Y is heavy-tailed, we would like to have an estimate for F_2 such that it gives an $(1 \pm \epsilon)$ approximation with probability (say) $1 - \frac{1}{n}$. This can be achieved by taking median of Y_1, Y_2, .., Y_s each an independent estimates of Y.

The sketch operator R can be viewed as an approximate embedding of l_2^n to sketch space $\mathcal{C}$ such that

1. Each point in $\mathcal{C}$ can be described using only a few number so $C \subset \mathbb{R}^t$, $t << n$, and
2. value of $l_2(S)$ is approximately equal to $F(C(S))$, where $F(C(S)) = F(Y_1, Y_2, ..Y_t) = \text{median}(Y_1, Y_2, .., Y_t)$

Note that F being a median operator (C, F) is not a normed space. So performing any nontrivial operations in the sketch space (e.g. clustering, similarity search, regression etc.) becomes difficult. Does there exist a dimensionality reduction technique via sketching for which the sketch space is a normed space. Below we see one such example, where sketching achieves dimensionality reduction from l_2^n to l_2^t, for $t = O(\frac{1}{\epsilon^2} \log n)$.

2 A Different Linear Sketch

In stead of having ±1 as the entries of sketch operator R, we let each of its entry be a i.i.d. gaussian random variables from $\mathcal{N}(0,1)$. Recap the properties of normal distribution.

Normal distribution $\mathcal{N}(0,1)$:

- Range $(-\infty,\infty)$
- Density $f(x)=e^{-x^2}/\sqrt{2\pi}$
- Mean=0, Variance=1

Basic facts

- If X and Y are independent random variables with normal distribution then so is $X+Y$
- If X and Y are independent with mean 0 then $\mathsf{E}\big[[X+Y]^2\big]=\mathsf{E}\big[X^2\big]+\mathsf{E}\big[Y^2\big]$
- $\mathsf{E}\big[cX\big]=c\mathsf{E}\big[X\big], \mathsf{Var}\big[cX\big]=c^2\mathsf{Var}\big[X\big]$

Therefore, our estimation procedure is as follows. We have $[Z_1,Z_2,..,Z_t]=Rx$ and let $Y=\sum_{i=1}^{t}Z_i^2$. We return Y/t

Let us consider any Z_i, We have $Z_i=\sum_{j=1}^{n}x_jr_j^i$, where r_j^i is the entry in the cell $R[i,j]$.

$$\begin{aligned}
\mathsf{E}\big[Z_i^2\big] &= E[(\sum_j r_j^i x_j)^2] \\
&= E[\sum_j (r_j^i)^2x_j^2 + 2\sum_{j<k}\mathsf{E}\big[x_jx_kr_j^ir_k^i\big] \\
&= \sum_j x_j^2\mathsf{E}\big[(r_j^i)^2\big] + 2\sum_{j<k}x_jx_k\mathsf{E}\big[r_j^i\big]\mathsf{E}\big[r_k^i\big] \\
&= ||x||_2^2
\end{aligned}$$

Here the third equality comes from considering that all r_j^i and r_k^i are independent and the fourth equality comes from the fact that $\mathsf{E}\big[r_j^i\big]=0$ for all r_j^i and $\mathsf{E}\big[(r_j^i)^2\big]=\mathsf{Var}\big[r_i^j\big]=1$.

Therefore, we have $\mathsf{E}\big[Y/t\big]=\frac{1}{t}\sum_i\mathsf{E}\big[Z_i^2\big]=||x||_2^2$.

We now want to show that Y concentrates around its expectation. For this we use JL lemma. From JL lemma, by setting $t=O(\frac{\log n}{\epsilon^2})$, there exist constant $C>0$ s.t. for small enough $\epsilon>0$

$$\Pr[Y-t\|x\|_2^2]>\epsilon^2t\|x\|_2^2\le e^{-C\epsilon^2t}$$

The above implies

$$\Pr[Y/t-\|x\|_2^2]>\epsilon^2\|x\|_2^2\le e^{-C\epsilon^2t}$$

We now prove the above inequality which also establishes JL lemma.

3 Johnson-Lindenstrauss Lemma

We shall assume without loss of generality $||x||_2^2 = 1$ (can be ensured by scaling). We defined $Z_i = \mathbf{r^i x} = \sum_{j=1}^n r_j^i x_j$ where r_j^i are i.i.d. random variables drawn from $\mathcal{N}(0,1)$. Let $Z = [Z_1, Z_2, ..., Z_t]$. We have $\mathsf{E}\big[Y\big] = \mathsf{E}\big[||Z||_2^2\big] = t$. We only prove the upper tail here. The proof of the lower tail is similar.

Lemma 1. $\mathsf{Pr}\big[||Z||_2^2 \geq t(1+\epsilon)^2\big] \leq e^{-t\epsilon^2 + O(t\epsilon^3)}$.

Proof. We have Y as the random variable $||Z||_2^2$ and let $\alpha = t(1+\epsilon)^2$. For every $s > 0$, we have

$$\begin{aligned} \mathsf{Pr}\big[Y > \alpha\big] &= \mathsf{Pr}\big[e^{sY} > e^{s\alpha}\big] \\ &\leq \frac{\mathsf{E}\big[e^{sY}\big]}{e^{s\alpha}} \text{ by Markov inequality} \end{aligned}$$

We now calculate the numerator.

$$\begin{aligned} \mathsf{E}\big[e^{sY}\big] &= \mathsf{E}\big[e^{s(Z_1^2+Z_2^2+...+Z_t^2)}\big] \\ &= \mathsf{E}[\prod_{i=1}^t e^{sZ_i^2}] \\ &= \prod_{i=1} \mathsf{E}\big[e^{sZ_i^2}\big] \text{ by independence of } Z_i^2 \end{aligned}$$

We now calculate $\mathsf{E}\big[e^{sZ_i^2}\big]$. We have $Z_i = \sum_{j=1}^n r_j^i x_j$ where each $r_j^i \sim \mathcal{N}(0,1)$. By 2-stability of normal distribution, Z_i is distributed as $||x||_2 G$ where $G \sim \mathcal{N}(0,1)$. Since $||x||_2 = 1$, $Z_i \sim \mathcal{N}(0,1)$. Hence,

$$\begin{aligned} \mathsf{E}\big[e^{sZ_i^2}\big] &= \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} e^{sa^2} e^{-a^2/2} da \\ &= \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} e^{(s-\frac{1}{2})a^2} da \end{aligned}$$

We now apply change of variables $u^2 = (1-s2)a^2$. Therefore, by elementary calculations, $da = \frac{1}{\sqrt{1-2s}} du$ and the above integral becomes

$$\frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \frac{1}{\sqrt{1-2s}} e^{-u^2/2} du = \frac{1}{\sqrt{1-2s}}$$

Therefore

$$\mathsf{E}\big[e^{sY}\big] = \prod_{i=1} \mathsf{E}\big[e^{sZ_i^2}\big] = \frac{1}{(1-2s)^{t/2}}.$$

Hence

$$\mathsf{Pr}\big[Y > \alpha\big] \leq \frac{1}{(1-2s)^{t/2} e^{s\alpha}}$$

Since the above inequality holds for all $s > 0$, it holds for $s = \frac{1}{2} - \frac{t}{2\alpha}$. Inserting this value for s we have

$$\mathsf{Pr}\big[Y > \alpha\big] \leq \left(\frac{t}{\alpha}\right)^{-t/2} e^{-\frac{\alpha}{2}\left(1-\frac{t}{\alpha}\right)} = e^{\frac{1}{2}(t-\alpha)}\left(\tfrac{t}{\alpha}\right)^{-t/2}.$$

Recall that $\alpha = t(1+\epsilon)^2$. Thus we have

$$\begin{aligned} e^{\frac{1}{2}(t-\alpha)\left(\frac{t}{\alpha}\right)^{-t/2}} &= e^{\frac{t}{2}(1-(1+\epsilon)^2)}e^{-\frac{t}{2}\ln\left(\frac{t}{\alpha}\right)} \\ &= e^{\frac{t}{2}(-2\epsilon-\epsilon^2-\ln\frac{1}{(1+\epsilon)^2})} = e^{t(-\epsilon-\epsilon^2/2+\ln(1+\epsilon))} \\ &= e^{t(-\epsilon-\epsilon^2/2+\epsilon-\epsilon^2/2+\epsilon^3/3-\ldots)} = e^{-t\epsilon^2+O(t\epsilon^3)} \end{aligned}$$

□

Lemma 2 (Johnson-Lindenstrauss)**.** *For any $0 < \epsilon < 1$ and any integer N, let t be a positive integer such that*

$$t \geq \left(\frac{4\ln N}{\frac{\epsilon^2}{2}+\frac{\epsilon^3}{3}}\right)$$

Then for any set V of N points in R^n, there is a map $f : R^n \to R^t$ such that for all u and $v \in V$.

$$(1-\epsilon)\|u-v\|_2^2 \leq \|f(u)-f(v)\|_2^2 \leq (1+\epsilon)\|u-v\|_2^2$$

Proof. Set f to be the sketching matrix $t \times n$ with each entry being i.i.d. drawn from $\mathcal{N}(0,1)$. Let $Z = f(u) - f(v)$ for a given u and v in V. Then by Lemma 1,$\mathsf{Pr}\left[(1-\epsilon)\|u-v\|_2^2 \leq \|f(u)-f(v)\|_2^2 \leq (1+\epsilon)\|u-v\|_2^2\right] \geq 1 - e^{-t\epsilon^2+O(t\epsilon^3)} \geq 1 - \frac{1}{N^3}$. Now there are at most $\binom{N}{2}$ pairs, hence by union bound for all $u, v \in V$,

$$\mathsf{Pr}\left[(1-\epsilon)\|u-v\|_2^2 \leq \|f(u)-f(v)\|_2^2 \leq (1+\epsilon)\|u-v\|_2^2\right] \geq 1 - e^{-t\epsilon^2+O(t\epsilon^3)} \geq 1 - \frac{1}{N}.$$

Therefore, by repeating a few times one can obtain the required map. □

JL lemma has found many applications is computer science. We now focus on one particular application of it namely, *n*earest neighbor search

4 Nearest Neighbor Problem

Given a set of points V, $|V| = N$, a distance metric d and a query point q, we want to find out the point $x \in V$ nearest to q. We first focus on the related near neighbor problem, where we ask given a set of points V, $|V| = N$ and a query point q, does there exist a point $x \in V$ such that $d(x,q) \leq R$. Clearly, if we can solve this later question, then with an addition of $O(\log N)$ increase in query time the nearest neighbor problem can be solved within arbitrary $(1+\epsilon)$ ($\epsilon > 0$ is a constant) approximation by binary search on R.

When dimension is low, both nearest neighbor problem and near neighbor problem can be solved efficiently. For example, when dimension (denoted by d) is 2, one requires space $O(N)$ and query time $O(\log N)$ to solve both of these problems. However, with increasing d, either the required space or the query time becomes exponential in d. This is known as *curse of dimensionality.* To tackle this, one resort to *approximate near neighbor problem* which we formally define below.

Definition 3 ((c, R)-Near Neighbor Problem)**.** *Given a set of points V, a distance metric d and a query point q, the (c, R)-Near Neighbor problem, $c \geq 1$, requires if there exists a point x such that $d(x, q) \leq R$, then one must find a point x' such that $d(x', q) \leq cR$ with probability $> (1 - \delta)$ for a given $\delta > 0$.*

We now introduce locality sensitive hashing (LSH) to solve (c, R)-Near Neighbor problem.

5 Locality Sensitive Hashing

The basic intuition behind LSH is that two points that are close to each other should hash to the same bucket with high probability, while those which are far apart should hash to different buckets. During preprocessing, we hash all the points in V to the respective buckets. When a query q comes, one only searches in the buckets that contain q.

Definition 4. LSH *A family of hash functions H is said to be (c, R, p_1, p_2)-sensitive for a distance metric d, when:*

1. $\Pr_{h \sim H}[h(x) = h(y)] \geq p_1$ *for all x and y such that $d(x.y) \leq R$*
2. $\Pr_{h \sim H}[h(x) = h(y)] \leq p_2$ *for all x and y such that $d(x.y) > cR$*

For H to be LSH *one must have $p_1 > p_2$.*

Example 5. *Let $V \subseteq [0, 1]^n$ and $d(x, y) =$ Hamming distance between x and y. Let $R << n$ and $cR << n$, define $\mathcal{H} = \{h_1, h_2, ..., h_n\}$ such that $h_i(x) = x_i$. $p_1 \geq 1 - \frac{R}{n}$ and $p_2 \leq 1 - \frac{cR}{n}$.*

6 Description of Algorithm:

Algorithm 1 Preprocessing

```
for all $x \in V$ do
  for all $j \in [L]$ do
    add $x$ to $bucket_j(g_j(x))$
  end for
end for
```

Algorithm 2 Query(q)

```
for $j = 1$ to $L$ do
  for all $x \in bucket_j(g_j(q))$ do
    if $d(x, q) \leq cR$ then
      return x
    end if
  end for
end for
return none
```

Let H be a family of LSH which is (c, R, p_1, p_2) -sensitive. Select $K \times L$ hash functions from this family independently and randomly: $h_{i,j} \sim H, i \in [1, K], j \in [1, L]$. Now define, $g_j = \langle h_{1,j}, h_{2,j}, ..., h_{K,j} \rangle$ for all $j \in [1, L]$. That is,

$$
\begin{aligned}
g_1 &= \langle h_{1,1}, h_{2,1} .....h_{k,1} \rangle \\
g_2 &= \langle h_{1,2}, h_{2,2} .....h_{k.2} \rangle \\
&\vdots \\
g_L &= \langle h_{1,L}, h_{2,L} .....h_{k,L} \rangle
\end{aligned}
$$

The preprocessing time is $O(NLK)$ assuming computation of each $h_{i,j}$ requires $O(1)$ time. Then computing L $g_1, g_2, ..., g_L$ require $O(KL)$ time and each of the N points need to be hashed giving the required time complexity.

Hashing query point q by $g_1, g_2, .., g_L$ require $O(KL)$ time. Suppose, F be the probability for any given j that a point x is hashed to the same bucket by g_j as q but $d(x, q) > cR$. Then the number of such points hashed to the same bucket at a particular level on expectation is NF. There are L levels, so searching through all these points require $O(NLF)$ time giving the said running time. Therefore, in order to obtain a good query time, we want F to be as small as possible.

References

[1] Piotr Indyk, Rajeev Motwani: Approximate Nearest Neighbors: Towards Removing the Curse of Dimensionality. STOC 1998: 604-613